Visualizing Webpage Changes Over Time (White Paper for NEH/IMLS DHAG HAA-256368-17)

Abigail Mabe, Dhruv Patel, Maheedhar Gunnam, Surbhi Shankar, Sawood Alam, Michael L. Nelson (Co-PI), Michele C. Weigle (PI)
Department of Computer Science
Old Dominion University
Norfolk, VA

Mat Kelly
College of Computing & Informatics
Drexel University
Philadelphia, PA

Deborah Kempe (Co-PI)
Frick Art Reference Library
New York Art Resources Consortium
New York, NY

Pamela Graham (Co-PI), Alex Thurman (Co-PI)
Columbia University Libraries
New York, NY

ABSTRACT

We report on the development of TMVis, a web service to provide visualizations of how individual webpages have changed over time. We leverage past research on summarizing collections of webpages with thumbnail-sized screenshots and on choosing a small number of representative past archived webpages from a large collection. We offer four visualizations: image grid, image slider, timeline, and animated GIF. Embed codes for the image grid and image slider can be produced to include these on separate webpages. The animated GIF can be downloaded as an image file for the same purpose. This tool can be used to allow scholars from various disciplines, as well as the general public, to explore the temporal nature of web archives. We hope that these visualizations will just be the beginning and will provide a starting point for others to expand these types of offerings for users of web archives.

1 INTRODUCTION

The web has become an integral part of our lives, shaping how we get news, shop, and communicate. The web is also dynamic; webpages that existed two years ago may not exist today [14, 15], webpages that exist today may not exist in two years, and even if webpages continue to exist, they can have significantly different content than before [10, 13]. Because of this, humanities scholars and social scientists are recognizing that web archives, such as the Internet Archive[1], are essential resources for their research [4, 18, 23, 24].

A goal of the Internet Archive is to save as much of the web as possible. Other organizations, including libraries and museums, are concerned with creating focused collections of archived webpages about particular topics, such as human rights[2] and art history[3]. Humanities scholars may be interested in obtaining an overview of the topic of a particular webpage in a collection or in observing how that webpage has changed over time. Many web archive interfaces provide only a textual list of archived versions, or *mementos*, of a webpage, which requires that each one be explored individually. This can be a tedious process as the number of mementos for each webpage grows.

As a first step to address this problem, we have developed visualizations that allow users to observe how a single webpage changes over time without the user accessing each memento individually. We do this by using the list of mementos of a webpage and choosing to display the screenshots, or thumbnails, of the most unique mementos, as a visual summarization. In addition to aiding researchers, this would also be useful in educating the general public about the temporal and dynamic nature of the web.[4]

2 VIEWING MEMENTOS OF A WEBPAGE OVER TIME

We use Memento [25] terminology, where URI-R identifies a web resource (or, original webpage), URI-M identifies that web resource at a particular point in time (or, a memento), and URI-T identifies its *TimeMap*, which is a list of all of the mementos (URI-Ms) for the original resource (URI-R). Having a visual representation, or thumbnail, of each of the mementos in a TimeMap is helpful in quickly determining both the overall focus of the webpage and also how the design of that webpage has changed over time. Other work on displaying webpage change has also included thumbnail views of individual mementos [1, 19, 22]. But a problem arises when there are too many thumbnails to display. For example, the Internet Archive has over 40,000 mementos for bbc.co.uk over a 20-year span. So even if the Internet Archive has thumbnails for all the mementos, some form of sampling will be necessary because the cognitive load of processing all the mementos will be beyond what the user can handle [17]. In addition, for webpages that do not change frequently, there may be many mementos with the same content, resulting in duplicate thumbnails.

In our previous work [3], we developed a technique to determine which mementos are useful to display. The algorithm compares the amount of change in the HTML of consecutive mementos and selects the most unique. This process is much more efficient, in

[1] https://web.archive.org
[2] https://hrwa.cul.columbia.edu/
[3] http://www.nyarc.org/content/web-archiving/

[4] This white paper has also been published as a tech report on the arXiv pre-print server [16]

both time and space, than generating all thumbnails and then performing image processing. In this work, we implement this technique for TimeMap visualization. For example, in Figure 1 we show thumbnails of 700 mementos of www.apple.com (we were unable to capture a figure showing all 8000 mementos) and a set of 69 thumbnails chosen by our algorithm from the full set of 8000. There are many duplicate thumbnails in Figure 1a that have been eliminated in Figure 1b.

3 VISUALIZATIONS

We have implemented four different visualizations to show how a webpage changes over time: image grid, image slider, timeline, and animated GIF. The initial implementations of the visualizations were developed by Shankar [20]. In addition to sampling over the full TimeMap, we allow users to select a time range over which the summarization should be generated. In all of these visualizations, the thumbnail summarization algorithm will choose the thumbnails to display. The visualizations are just different ways to present and interact with the same information.

3.1 Image Grid

The Image Grid shows the thumbnails of all the unique mementos arranged in a left to right, top to bottom manner. Clicking on the thumbnail loads the source memento, giving the user the opportunity to explore further. On the top right of each thumbnail in the grid is a refresh button to allow users to regenerate the thumbnail if it appears incomplete and an 'X' to allow users to remove thumbnails from the visualizations. Figure 2 shows an example of the grid view for http://www.odu.edu/.

3.2 Image Slider

The Image Slider imitates the photo slider functionality used in Apple's iPhoto[5]. By moving the cursor across the thumbnail image, the next thumbnail is displayed. As with the Image Grid, clicking on the thumbnail loads the source memento. The user can cycle through the thumbnails by clicking arrow buttons to the left and right of the slider. Figure 3 shows a static example of the image slider for http://columbia.edu/cu/english/.

3.3 Timeline

The Timeline view arranges the thumbnails according to the mementos' datetimes (time of capture). The Timeline view is equipped with zoom, next, previous, next unique, previous unique buttons to allow the user to easily navigate between the unique and regular mementos. The unique mementos are represented with yellow stripes and the regular ones are represented by gray stripes on the timeline. Non-unique mementos do not have a screenshot, but are represented by the thumbnail of the previous unique memento with an indication that this memento is "similar to" that memento. The Timeline view is based on Timeline Setter library[6] [21], developed by ProPublica. Figure 4 shows an example of the Timeline view for http://www.odu.edu.

[5]http://web.archive.org/web/20150101033528/http://apple.com/mac/iphoto/
[6]http://propublica.github.io/timeline-setter/

3.4 Animated GIF

The Animated GIF visualization, shown in Figure 5, takes the thumbnails from the Image Grid and uses the GifShot library [28] to create an animated GIF. This GIF can be downloaded by clicking the "Download GIF" button. The user is given the option to include a timestamp on each screenshot in the GIF. When selected, a watermark of the appropriate datetime is added to each screenshot. This visualization also allows the user to adjust the time interval in seconds between each frame of the animated GIF. If any settings are changed, the GIF will be updated once the user presses the "Update GIF" button.

4 SELECTING UNIQUE THUMBNAILS

We select the most unique thumbnails based on a comparison of the SimHash [6] of each memento's HTML source. To improve the response time for very large TimeMaps (over 1000 mementos), we first sample a maximum of 1000 mementos before computing the SimHash and continuing the selection algorithm. These techniques are described further below. The initial implementation of the base summarization algorithm was developed by Kelly [11].

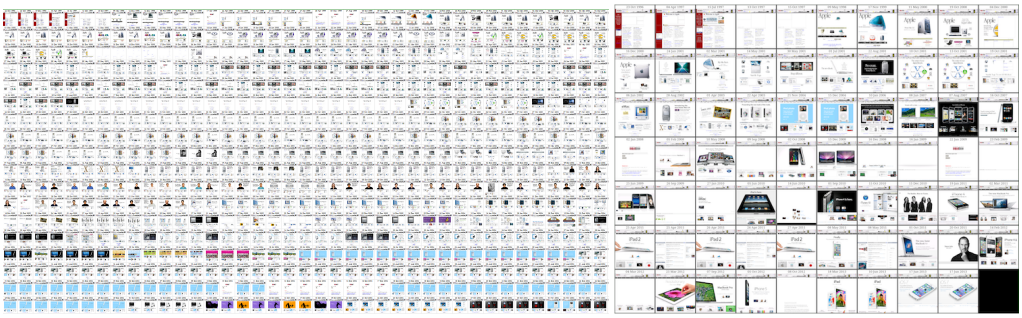
4.1 SimHash

AlSum summarization [3] relies on comparing the HTML source of mementos rather than images of the rendered webpage. Downloading and analyzing just the HTML source is much more efficient in both time and storage than rendering the entire webpage, taking a screenshot, and comparing images. Because the rendering process is expensive, we only want to render and take screenshots of those mementos we deem to be most unique. To do this, we download the HTML content of all mementos in the TimeMap and compute the SimHash of each memento. Then we compare the SimHashes of two mementos at a time, computing the Hamming distance [8] between the two SimHashes. We can use different Hamming distance thresholds to require different amounts of changes and thus produce different numbers of representative thumbnails.

The process of choosing the best thumbnails using SimHash is outlined in Figure 6. The filtering process begins by selecting the first memento in the TimeMap to be included in the summarization. This memento will act as a baseline to compare to subsequent mementos. In Figure 6, the Hamming distance threshold is denoted as HDT, and each memento is M1, M2, ... Mn. The first memento, M1, is compared to each consecutive memento until the Hamming distance is greater than or equal to the threshold. In the diagram, this occurs when M1 is compared to M3.

To demonstrate why SimHash is a better measure in calculating the webpage similarities than other hash techniques like MD5, we consider three mementos, M1, M2, and M3, from the TimeMap of http://www.odu.edu. The thumbnails are shown in Figure 7. The thumbnails for mementos M1 and M2 look similar, but M3 is distinct from the other two. Upon closer look, M1 and M2 contain different content, though the layout is the same. First, we compute the MD5 hashes of the HTML content of these mementos:

```
$ curl URI-M1 | md5sum -> fc8e53aebb9061f390aba82665581295
$ curl URI-M2 | md5sum -> d546e192eab633f4d1b4451399c8adcc
$ curl URI-M3 | md5sum -> 5e98bc5367c86f3ffaea0b8c3deb3f5d
```



(a) 700 thumbnails

(b) 69 thumbnails, sampled by our algorithm

Figure 1: Thumbnails from the www.apple.com TimeMap

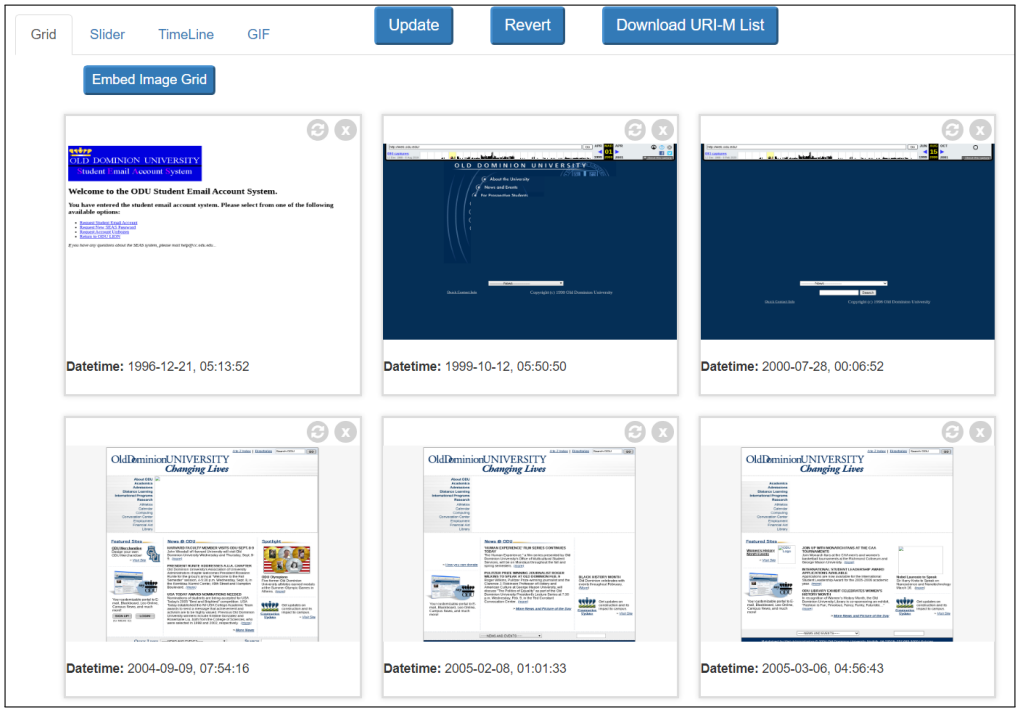


Figure 2: Image Grid for http://www.odu.edu/

After the hashes are generated, the Hamming distance is calculated between the pairs (M1, M2) and (M2, M3). The Hamming distance is the minimum number of substitutions required to convert one string to another. The results are as shown below:

```
$ node hammingdistance M1 M2 -> 30
$ node hammingdistance M2 M3 -> 32
```

For MD5, we see that the distances between (M1, M2) and (M2, M3) are similar, meaning that MD5 considers them to have similar amounts of difference.

Now we repeat the same process using SimHash as the hashing technique:

```
$ curl URI-M1 | SimHash -> 8c27981eaed151cfa645ad823932eac6
$ curl URI-M2 | SimHash -> 8c27981faad951cf8645ad823d32eac2
$ curl URI-M3 | SimHash -> fa3799170258494b9443b9be3977a84e
```

The Hamming distances are calculated as below:

```
$ node hammingdistance M1 M2 -> 6
$ node hammingdistance M2 M3 -> 27
```

Here we see that with SimHash, the distance between (M1, M2) is much smaller than the distance between (M2, M3). This reflects the

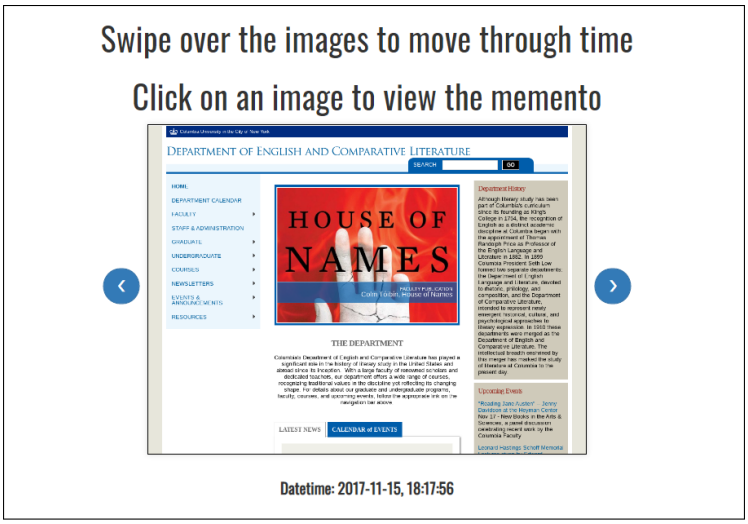


Figure 3: Image Slider for http://columbia.edu/cu/english/

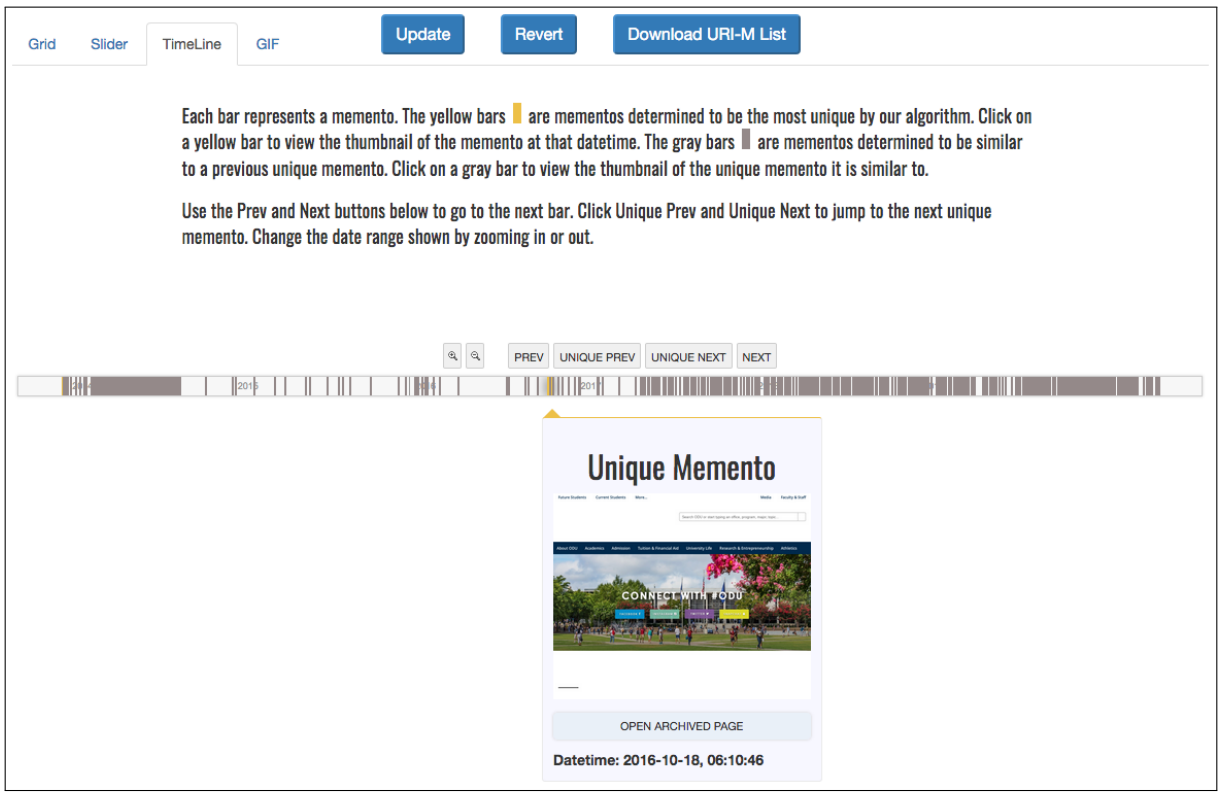


Figure 4: Timeline view for http://www.odu.edu/

notion that we have after observing the screenshots of the rendered pages. Hence we can conclude that SimHash correlates better with the webpage similarity.

4.2 TimeMap Sampling Algorithm

Some TimeMaps contain hundreds of thousands of mementos. To efficiently calculate the most representative mementos from such large TimeMaps, we take a sample of up to 1000 mementos from the TimeMap to analyze. The algorithm used to extract the sample

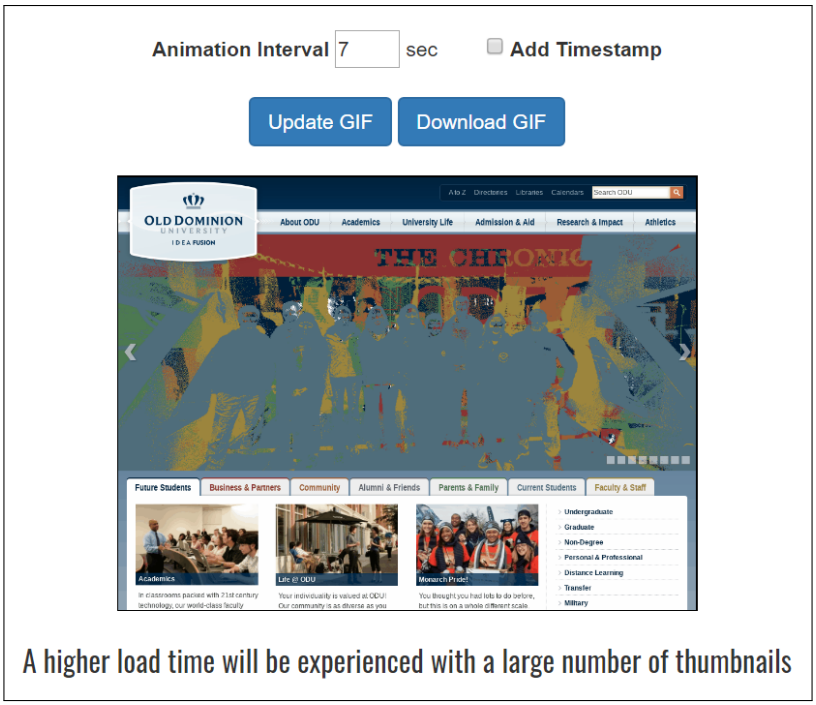


Figure 5: Animated GIF for http://www.odu.edu/

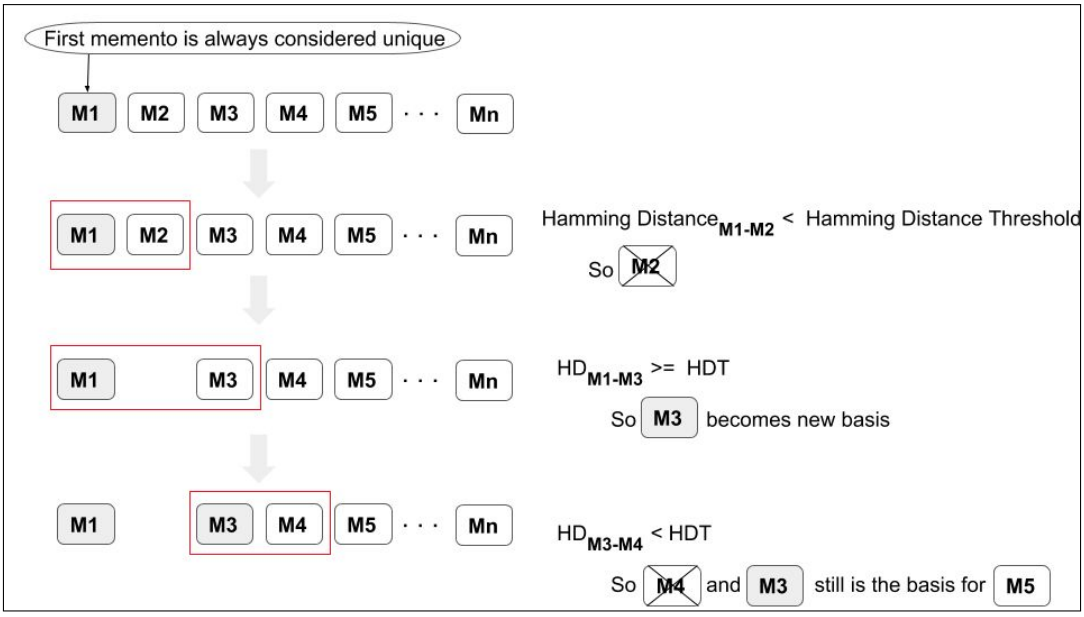
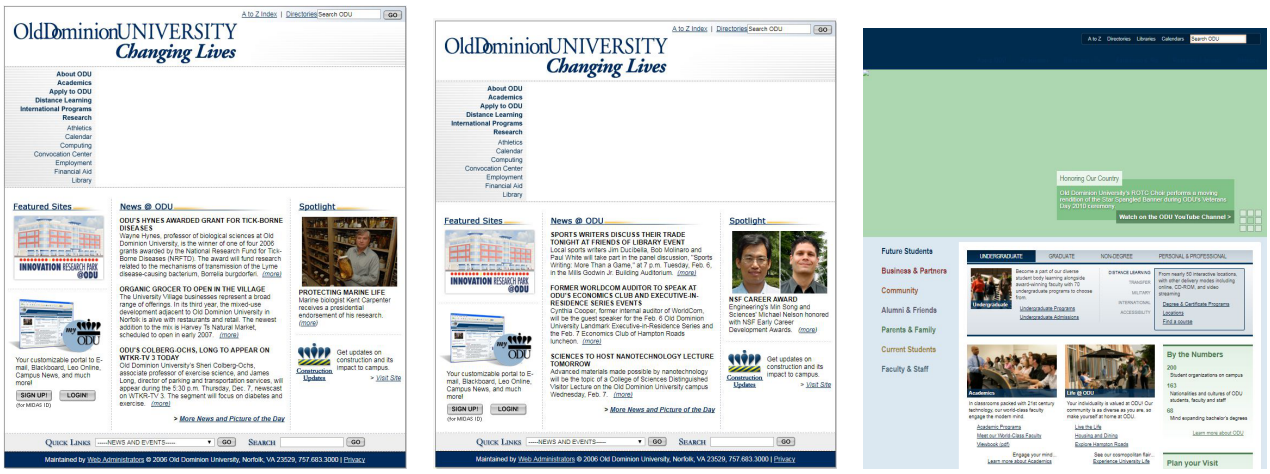


Figure 6: Determining the most representative mementos using the Hamming distance

of mementos is illustrated in Figure 8. The TimeMap is split into 250 equal partitions. For example, if the TimeMap contains 2000 mementos, each partition will contain 8 mementos. From each partition, up to four mementos will be chosen. The algorithm always chooses the first memento in the set. Each time a memento is chosen, a counter of the number of mementos left that can be selected from the partition is decremented. The datetime of the first memento is then compared to the datetime of the second memento. If the time between the mementos is less than 3 days, the second memento will be skipped. The first memento is compared to each consecutive



(a) M1 (b) M2 M3

Figure 7: Thumbnails of M1, M2, M3 from the SimHash example

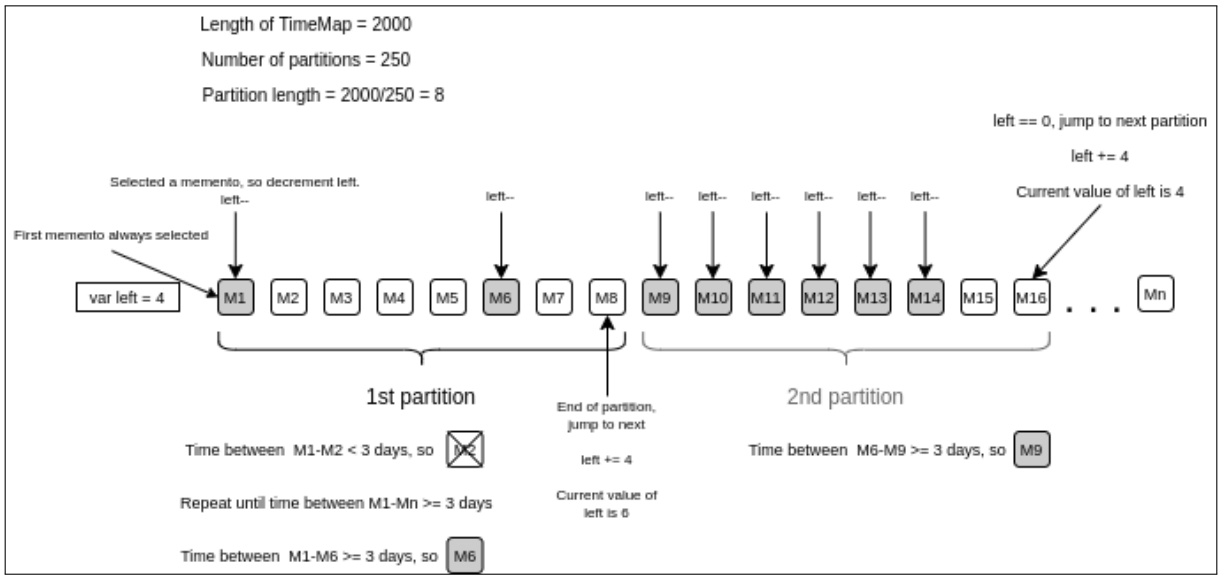


Figure 8: TimeMap Sampling Algorithm

memento until the distance between datetimes is at least 3 days. As shown in Figure 8, this occurs when the first memento is compared to the sixth memento. If the end of the partition is reached before the counter is 0, then four mementos plus the leftover mementos can be chosen from the next partition. In Figure 8, only two mementos were selected from the first partition, so six mementos can be selected from the second partition. This process continues until the end of the TimeMap is reached.

The algorithm is designed to extract a sample that represents the distribution of the original TimeMap. For example, if most of the mementos in a particular TimeMap are from 2016, then most of the mementos in the sample will also be from 2016. Figure 9 shows a histogram of the TimeMap for moma.org compared to a histogram of the filtered TimeMap. The histograms are binned by month and year of the memento datetime. It can be seen from the histograms that the filtered TimeMap has proportions similar to the original TimeMap. Sections of the TimeMap where hundreds of mementos are from the same month are omitted. This allows the unique mementos to be calculated more quickly since hundreds of similar mementos do not have to be compared.

5 CACHING TIMEMAPS AND THUMBNAILS

The determination of which thumbnails to choose as the most representative for a particular TimeMap may change over time, as new mementos are archived. For efficiency, we do not want to recompute SimHash values for mementos we have already processed,

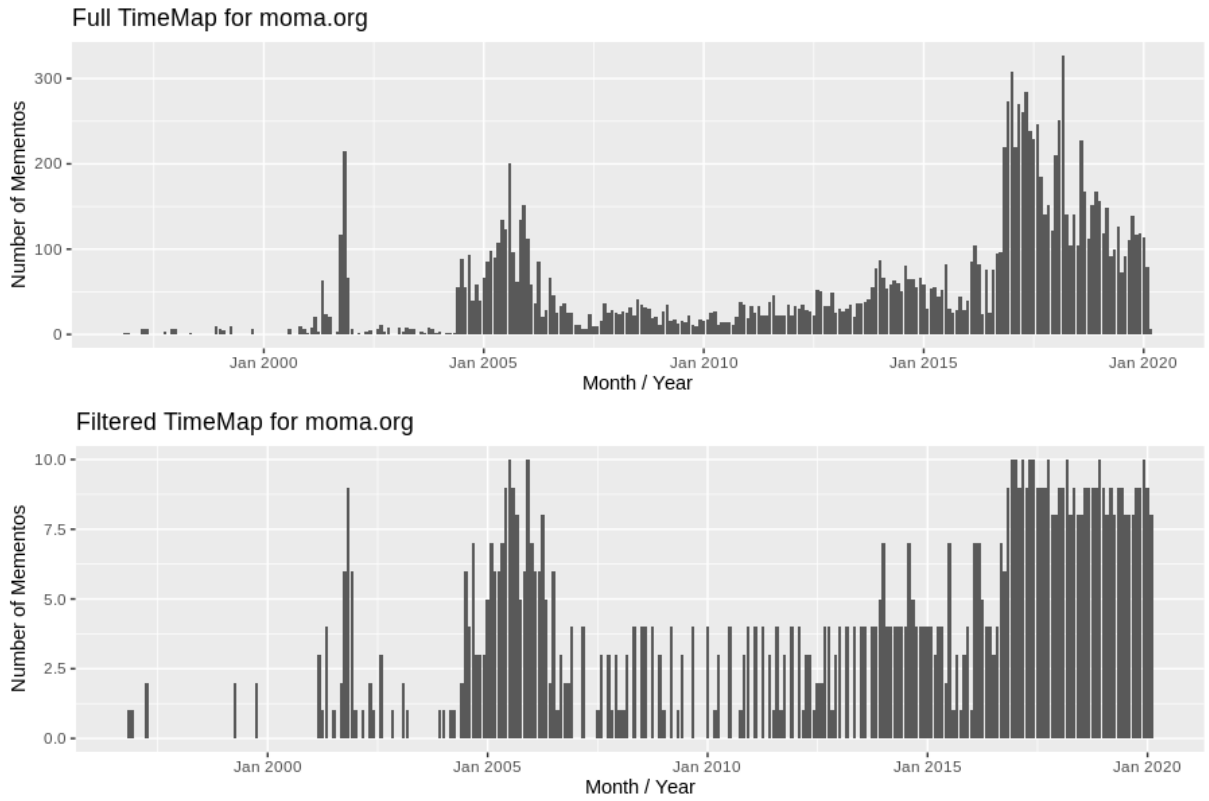


Figure 9: Sample selected from http://moma.org/

nor do we want to re-render thumbnails when those have already been generated. Thus, we have developed a process for caching TimeMaps, SimHash values, and thumbnail images.

5.1 TimeMap Caching

As the web archiving process continues, a TimeMap for a webpage is likely to continue to grow. In addition, sometimes archives might even delete a memento from a TimeMap either temporarily or permanently. We have accounted for these two instances by checking if the number of mementos we have cached for a particular URI-R matches the number of mementos the archive has for that URI-R. In order to ensure that the user is shown the most up to date information, we have developed the caching system shown in Figure 10 that stores current thumbnail summary information and updates the cache files as needed.

When the user inputs a URI-R for processing, the full TimeMap is fetched and stored in a cache file. This cache file stores an array of datetimes for each memento in the TimeMap. The naming convention of this file is `histogram_[archive name][collection identifier][URI-R]`. The user will then select the date range of mementos that they would like to view. After selecting a date range, the system checks if a SimHash cache file exists for the URI-R. If the file does not exist, the TimeMap is fetched and a new TimeMap object containing all of the mementos in the TimeMap is created. The mementos in the TimeMap object are then filtered down to consist of only mementos in the user-requested date range. If there are more than 1000 mementos after filtering down to the date range, the mementos are filtered again with the TimeMap sampling algorithm (described earlier) so that a sample of up to 1000 mementos remains. The SimHashes for the mementos in this sample will then be calculated and cached.

If the cache file does already exist for the specified URI-R, the file contents are read into a new TimeMap object. Since the mementos are from a previously fetched TimeMap, the cached mementos must be compared to the current TimeMap. To do this, the mementos in the histogram cache file are read into an array. This array is filtered, by both date range and with the TimeMap sampling algorithm, along with the mementos in the TimeMap object created from the previous file contents. If the TimeMap has remained unchanged since the previous SimHash cache file was created, then the mementos in the histogram cache array and the TimeMap object will be the same and execution may continue. If the histogram array contains more mementos than the TimeMap object, then the "Update Cache" process begins. Conversely, if the histogram array contains fewer mementos than the TimeMap object, then some mementos have been removed from the archive. These mementos must also be removed from the TimeMap object as the system may not be able to take a screenshot of a memento that no longer exists. Here the "Delete Extra Mementos" process begins (as described below). The purpose of this is to reduce the need for recomputing the

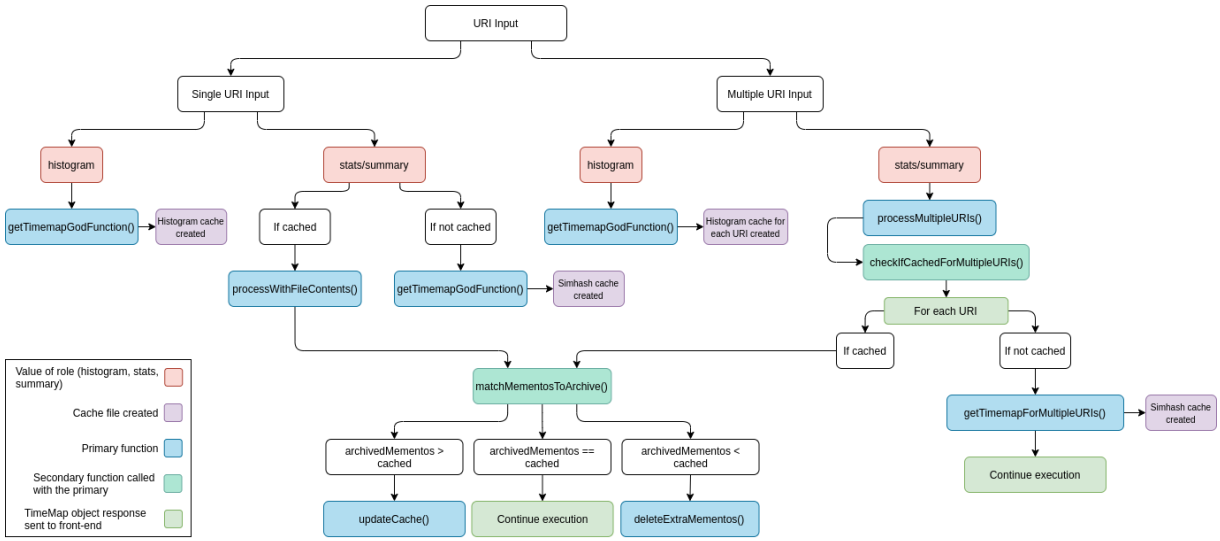


Figure 10: Cache File System for TMVis

SimHashes. Iterating through the memento lists can be completed in a matter of seconds. However, recomputing the SimHashes can take several minutes.

5.2 Update Cache Process

The "Update Cache" process updates the existing SimHash cache file for a given URI-R. This process starts by fetching the URI-R's TimeMap from the archive. A TimeMap object reads this data and filters mementos for date range if needed. If the number of mementos is greater than 1000, they are filtered down to include only a sample of 1000. The mementos that exist in the cache file are deleted from the object and stored in another object for later use. The SimHashes of all the mementos not in the cache file are calculated and then merged into the object containing the old cached mementos. This object is then sorted old to new by date and the cache file is overwritten with these mementos.

5.3 Delete Extra Mementos

The "Delete Extra Mementos" process is triggered when the number of cached mementos exceeds the number of mementos currently in the TimeMap. The datetimes of the mementos in the cached mementos array are compared to those in the archived mementos array. If a datetime from the cached array does not match a datetime in the archived array, the memento in the cached array is considered to be currently removed from the archive and therefore discarded from the array. After removing the cached memento from the array, the loop continues comparing the datetimes. The discarded mementos are not deleted from the cache file since they may be added back to the archive in the future. By the end of this process, the cached memento list and the archived memento list will be identical.

6 SYSTEM WALKTHROUGH

We have developed a web service that allows general users to view an overview of any TimeMap with the developed visualizations. The initial implementation of the web service was developed by Gunnam [7]. We have also developed embeddable versions (with simple embed scripts like those provided by Twitter, YouTube, etc.) that allows webpage authors to include these visualizations in their own webpages. The web service is available at http://tmvis.cs.odu.edu/, the source code is hosted on GitHub at https://github.com/oduwsdl/tmvis.

6.1 TimeMap Selection

From a user's perspective, there are three phases needed to summarize the URI. Figure 11 shows the first phase, the home page of the application where the user can input the URI-R, URI-M, or the URI-T. Again, according to Memento [25] terminology, URI-R identifies an original webpage, URI-M identifies a memento, and URI-T identifies a TimeMap. The user can choose the web archive to be used, either the Internet Archive or Archive-It, a collection and subscription-based archival service operated by the Internet Archive. Opting for Archive-It allows the user to include a collection number as an additional parameter. If no collection number is input, the value "all" is considered to be the collection number. We currently only support the Internet Archive and Archive-It, but the list of archives could be expanded to any Memento-compatible public web archive or Memento aggregator [2, 5].

The second phase in the URI summarization process begins with the user clicking the "View TimeMap" button after providing the proper input. The TimeMap is then requested from the appropriate archive and cached for future reference. The user is presented with a histogram displaying the number of mementos available over time as shown in Figure 12. The user may use the histogram to select a date range in the TimeMap to summarize, or they may choose to summarize the entire TimeMap. To choose a date range, the user may either click and drag over the histogram to select the desired range, or they can type the date range into the boxes below.

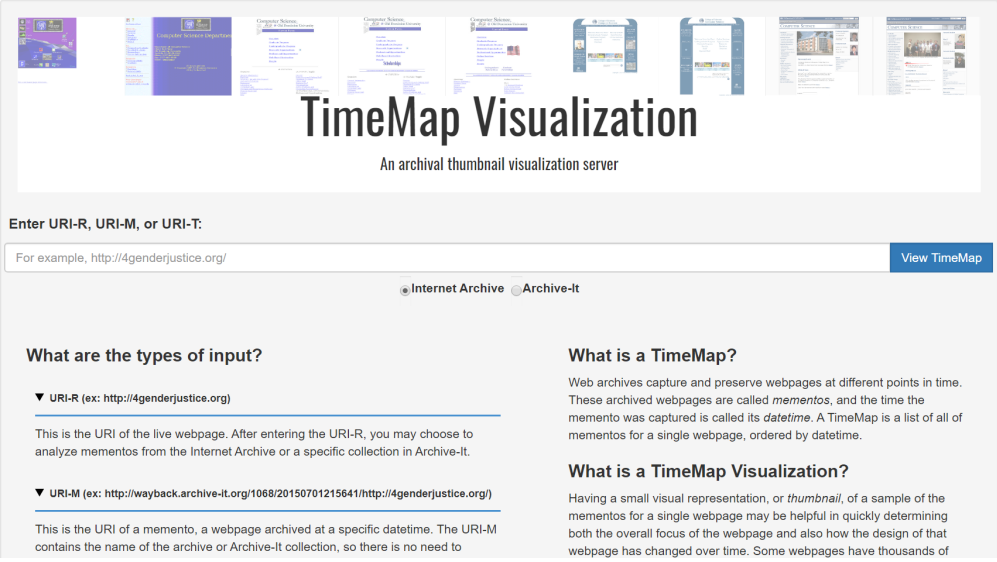


Figure 11: Home page of the service http://tmvis.cs.odu.edu/

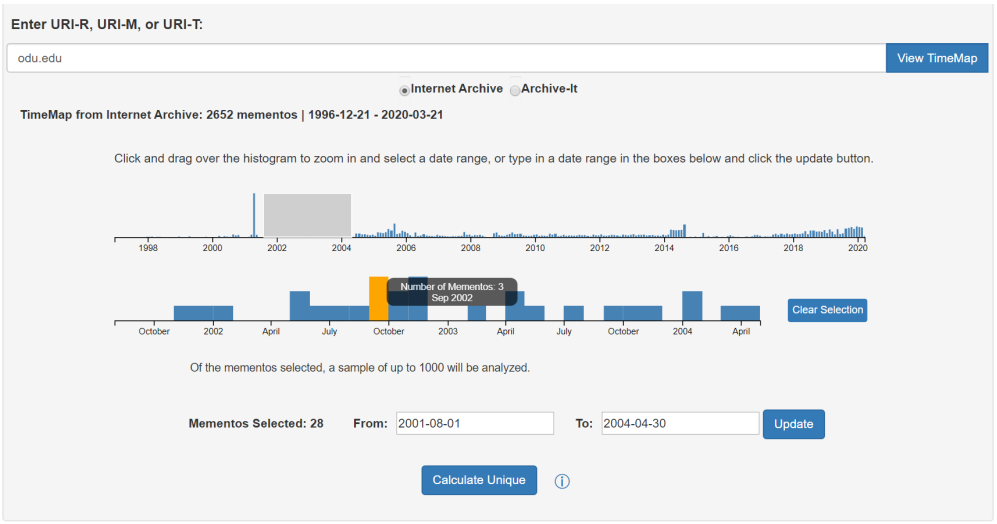


Figure 12: Histogram page of the service http://tmvis.cs.odu.edu

Some URIs have very few mementos and so users may be interested in viewing all of these mementos. Since processing just a few mementos can be done rather quickly, for URIs with less than 12 mementos, the user is given the option to process and display thumbnails for all mementos by clicking the "Generate All Thumbnails" button.

6.2 Comparing Mementos

When the user clicks the "Calculate Unique" button, the web service downloads the HTML source and computes the SimHash of the requested mementos. Then the SimHashes of the mementos are compared according to AlSum et al. [3]. The user is then notified with the continuous stream of events taking place at the server through a progress window, as shown in Figure 13.

Once the server side computation of unique number of representative thumbnails is complete, the user is presented with the date range of mementos under consideration along with options for choosing the number of representative thumbnails. Figure 14 shows the options available after processing http://odu.edu/ with the Internet Archive as the source. Figure 14 also shows that the TimeMap of http://odu.edu can be summarized using 127, 62, 51, 36, 32, 27, 26, 25, or 3 thumbnails. Choosing 127 thumbnails will summarize the TimeMap with more similar webpages and hence takes more time to generate. Choosing 3 thumbnails summarizes

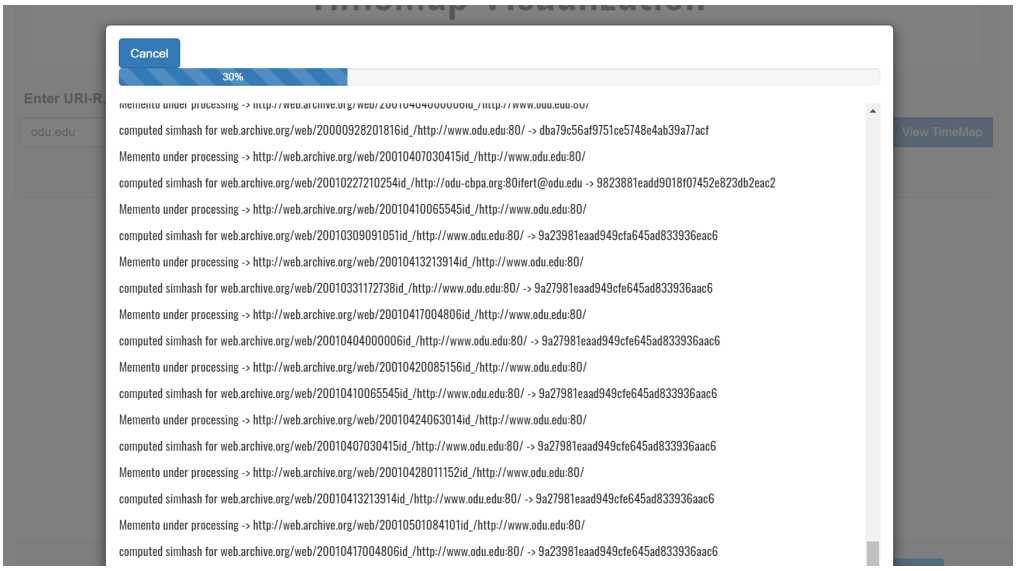


Figure 13: Home page of http://tmvis.cs.odu.edu with progress window

the TimeMap with more distinct webpages and takes a shorter amount of time to generate.

Once the representative memento filtering based on Hamming distance is complete, if the smallest Hamming distance produces a number of representative mementos greater than 9, then the option of three unique mementos will be given to the user in addition to the other options as shown in Figure 14. These three mementos are the first, last, and central mementos from the smallest list of unique mementos that was obtained from the smallest Hamming distance threshold. If the list of unique mementos is even and there are two mementos in the center of the array, the central memento is considered to be the one with the lower index.

6.3 Thumbnail Generation and Visualization

The third and final phase of thumbnail generation starts when the user chooses the desired number of unique thumbnails and then clicks on the "Generate Thumbnails" button. The web service uses Puppeteer[7], a headless web browser, to render each memento and capture a thumbnail screenshot. Once the thumbnails are captured for all the mementos, they are presented to the user with the four visualization widgets shown in Figures 2-5. The default tab shows the Image Grid, however users can switch between tabs to view the other visualizations: Image Slider, Timeline, and Animated GIF.

6.3.1 User Control Over Representative Thumbnail Selection. The Image Grid visualization tab allows the user to customize the thumbnails used for all of the visualizations. The user may remove thumbnails and refresh thumbnails. To remove a thumbnail, the user clicks the 'X' button on the top right corner of each thumbnail they would like removed. When clicked, the thumbnail becomes grayed out, showing that it is marked for removal. The user may click the button again to deselect the thumbnail. Once the user has selected which thumbnails they would like excluded from the grid view, the user can click the "Update" button at the top of the Image Grid to apply the changes. If the user would like to add back the thumbnails, the user can click the "Revert" button at the top of the Image Grid to revert back to the original set of thumbnails. Figure 15 shows an example of the Image Grid for http://odu.edu/ with three thumbnails selected for removal and the "Revert" and "Update" buttons at the top of the grid.

Mementos are loaded from web archives. The time to render a full webpage varies due to the number and types of embedded resources in the webpage and current load on the web archive server. Some mementos may take a long time to fully render, making it possible for the screenshot to be taken before all visual content appears on the page. This issue is accounted for by providing the user with the capability to request a new screenshot to be taken of a particular memento. This feature is incorporated into the user interface by placing a refresh icon in the top right corner of each memento in the Image Grid visualization, as shown in Figure 15. When a request is made for a screenshot to be retaken, the system waits for a longer time period than the prior attempt to allow the visual content to appear on the page. A new thumbnail is then generated from this new screenshot and each visualization is updated to include the new thumbnail.

6.3.2 Download URI-M List. Once the list of representative mementos has been determined, this list is written to a text file. The user can download this file by clicking on the "Download URI-M List" button on the Image Grid tab. This text file may be used for many purposes, including as input to Raintale [9], a tool that allows users to create and share stories using archived webpages.

6.3.3 Downloadable Animated GIF. The Animated GIF visualization, which has already been described (Figure 5), is easily embeddable into other webpages. We have provided a "Download GIF" button that allows the user to download the GIF image file. The user can customize the animated GIF in several ways. The user is given

[7] https://developers.google.com/web/tools/puppeteer

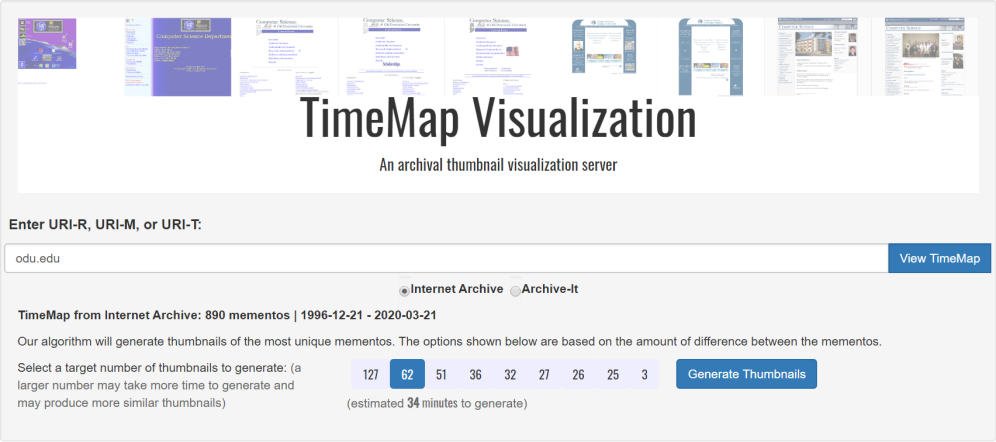


Figure 14: tmvis.cs.odu.edu showing the number of unique thumbnails calculated for http://odu.edu/

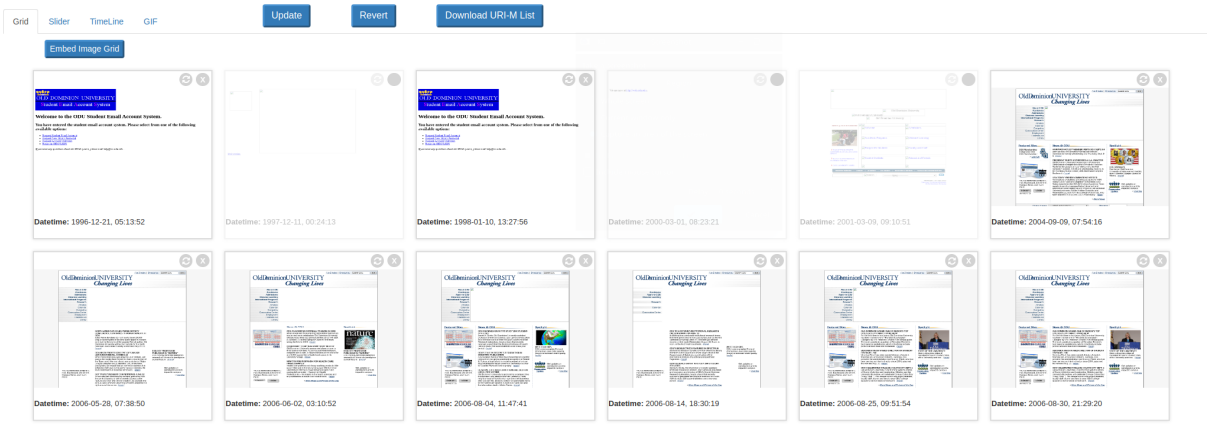


Figure 15: Demonstrating thumbnails selected for removal

the option to include a timestamp on each screenshot in the GIF. When selected, a watermark of the appropriate datetime is added to each screenshot and then the new Animated GIF is generated with the watermarked images. This visualization also allows the user to adjust the time interval in seconds between each frame of the animated GIF. If any settings are changed, the GIF will be updated once the user presses the "Update GIF" button.

6.3.4 Embed Feature for Image Grid and Slider. The Image Grid and Image Slider visualizations generated by TMVis give the user the option to embed the visualizations into other webpages. Users can click the "Embed Image Grid" and "Embed Image Slider" buttons to generate the embed code. Generating the embed code takes into account the user control over representative thumbnail selection by only including thumbnails that are currently being shown. This allows users to control which thumbnails they are sharing when they embed the visualization. The preferred visualization is encoded in an iframe with the selected thumbnails. For the Image Grid embed content, the 'X' and refresh are removed. An example of embedding the Image Grid, Image Slider, and Animated GIF in a simple webpage is provided at https://ws-dl.cs.odu.edu/vis/tmvis/embed-examples.html and shown in Figure 16.

6.4 Multiple URI Input

Some websites update their hostname, creating multiple URI-Rs that point to the same effective webpage. When this happens, any new mementos that are created after the hostname change may not correlate to the old hostname in the archive. Thus, there will be two separate TimeMaps. To accommodate this, users may input comma-separated URI-Rs. The system will fetch the TimeMap for each URI and merge them together, creating one TimeMap to process. This allows users to input all hostnames for a website and view all mementos associated with that particular website over time. For example, the user can input http://columbia.edu/cu/english/, http://english.columbia.edu/ as shown in Figure 17.

When "View TimeMap" is clicked, the system checks for a cache file for either URI-R. If one is not cached, then the TimeMap for the URI-R is fetched. The SimHashes for that URI-R are calculated and cached for future use. If none of the URI-Rs are cached, this process is completed for each URI-R individually. For any URI-R

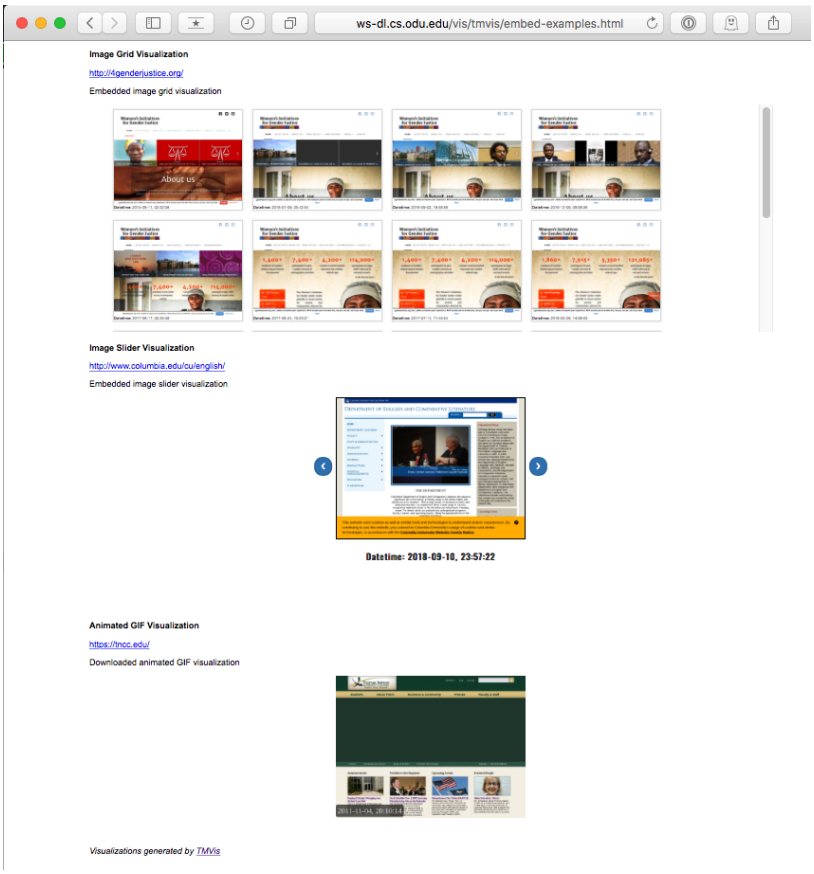


Figure 16: Visualizations embedded in another webpage

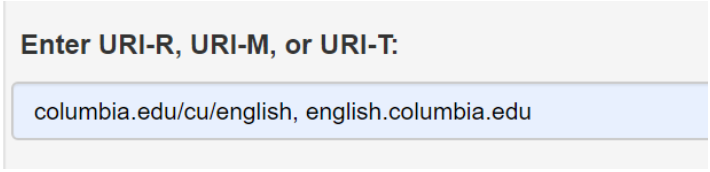


Figure 17: Multiple URI-R input

that is cached, the cache file contents are read into the TimeMap object. Once all the mementos for each URI-R are merged, they are sorted by date.

When the user loads the visualizations, the Image Grid and Animated GIF each have a URI stamp feature, as shown in Figure 18. The URI stamp is so that the user knows exactly which URI-R each particular memento came from. On the animated GIF, the URI stamp is optional. It is not optional on the Image Grid.

7 CONCLUSION

We have presented a description of the implementation of TMVis, a web service for producing four visualizations of how a webpage changes through time. To do this we compare the difference in the SimHashes of the HTML source of pairs of mementos to determine which mementos are the most unique. Then we render and capture thumbnail-sized screenshots of the chosen mementos. These are then displayed as Image Grid, Image Slider, Timeline, and Animated GIF visualizations. The Image Grid, Image Slider, and Animated GIF visualizations can be embedded in other webpages. The web service is available at http://tmvis.cs.odu.edu/, and the source code is hosted on GitHub at https://github.com/oduwsdl/tmvis.

ACKNOWLEDGEMENTS

This work has been supported by a NEH/IMLS Digital Humanities Advancement Grant (HAA-256368-17) [26]. We are grateful for the support of the National Endowment for the Humanities (NEH) and the Institute of Museum and Library Services (IMLS). This project is an extension of AlSum and Nelson's "Thumbnail Summarization

(a) URI-stamped Image Grid

(b) URI-stamped animated GIF

Figure 18: Visualizations with URI stamp functionality

Techniques for Web Archives" [3] and our previous work, funded by an incentive grant from Columbia University Libraries and the Andrew W. Mellon Foundation [12, 27].

REFERENCES

[1] Eytan Adar, Mira Dontcheva, James Fogarty, and Daniel S. Weld. 2008. Zoetrope: Interacting with the ephemeral web. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. 239–248. https://doi.org/10.1145/1449715.1449756

[2] Sawood Alam and Michael L. Nelson. 2016. MemGator - A Portable Concurrent Memento Aggregator. In *Proceedings of the 16th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '16)*. 243–244. https://doi.org/10.1145/2910896.2925452

[3] Ahmed AlSum and Michael L. Nelson. 2014. Thumbnail Summarization Techniques for Web Archives. In *Proceedings of the European Conference on Information Retrieval (ECIR)*. Amsterdam, 299–310.

[4] Sanjay K. Arora, Yin Li, Jan Youtie, and Philip Shapira. 2016. Using the Wayback Machine to mine websites in the social sciences: A methodological resource. *Journal of the Association for Information Science and Technology* 67, 8 (Aug. 2016), 1904–1915. https://doi.org/10.1002/asi.23503

[5] Luydmila Balakireva, Harihar Shankar, Martin Klein, Ilya Kremer, James Powell, and Herbert Van de Sompel. 2015. Time Travel APIs. http://timetravel.mementoweb.org/guide/api/.

[6] Moses S. Charikar. 2002. Similarity Estimation Techniques from Rounding Algorithms. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*. 380–388.

[7] Maheedhar Gunnam. 2018. How I Changed Over Time: A webservice to summarize TimeMaps based on SimHashed HTML content. ODU CS Masters Project Report. http://www.cs.odu.edu/~mweigle/papers/gunnam-ms-proj-18.pdf.

[8] Richard W. Hamming. 1950. Error detecting and error correcting codes. *The Bell System Technical Journal* 29, 2 (1950), 147–160.

[9] Shawn M. Jones. 2019. Raintale. https://ws-dl.blogspot.com/2019/07/2019-07-11-raintale-storytelling-tool.html, https://github.com/oduwsdl/raintale.

[10] Shawn M. Jones, Herbert Van de Sompel, Harihar Shankar, Martin Klein, Richard Tobin, and Claire Grover. 2016. Scholarly Context Adrift: Three out of Four URI References Lead to Changed Content. *PLoS ONE* 11, 12 (2016). https://doi.org/10.1371/journal.pone.0167475

[11] Mat Kelly. 2017. ArchiveThumbnails. https://github.com/machawk1/ArchiveThumbnails.

[12] Mat Kelly, Michael L. Nelson, and Michele C. Weigle. 2015. Visualizing Digital Collections of Web Archives (slides). http://www.slideshare.net/matkelly01/visualizing-digital-collections-of-web-archives-from-columbia-web-archiving-collaboration-conference.

[13] Martin Klein and Michael L. Nelson. 2008. Revisiting Lexical Signatures to (Re-)Discover Web Pages. In *Proceedings of the 12th European Conference on Research and Advanced Technology for Digital Libraries*. 371–382.

[14] Martin Klein, Herbert Van de Sompel, Robert Sanderson, Harihar Shankar, Lyudmila Balakireva, Ke Zhou, and Richard Tobin. 2014. Scholarly Context Not Found: One in Five Articles Suffers from Reference Rot. *PLoS ONE* 9, 12 (2014). https://doi.org/10.1371/journal.pone.0115253

[15] L. Lim, M. Wang, S. Padmanabhan, J. Vitter, and R. Agarwal. 2001. Characterizing web document change. *Advances in Web-Age Information Management* (2001), 133–144.

[16] Abigail Mabe, Dhruv Patel, Maheedhar Gunnam, Surbhi Shankar, Mat Kelly, Sawood Alam, Michael L. Nelson, and Michele C. Weigle. 2020. *Visualizing Webpage Changes Over Time*. Technical Report arXiv:2006.02487. https://arxiv.org/abs/2006.02487

[17] Richard E Mayer and Roxana Moreno. 2003. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist* 38, 1 (2003), 43–52.

[18] Ian Milligan. 2016. Lost in the Infinite Archive: The Promise and Pitfalls of Web Archives. *International Journal of Humanities and Arts Computing* 10, 1 (2016), 78–94. https://doi.org/10.3366/ijhac.2016.0161

[19] Kalpesh Padia, Yasmin AlNoamany, and Michele C. Weigle. 2012. Visualizing Digital Collections at Archive-It. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. Washington, DC, 15–18. https://doi.org/10.1145/2232817.2232821

[20] Surbhi Shankar. 2017. Visualizing Thumbnails Of Archived Web Pages. ODU CS Masters Project Report. http://www.cs.odu.edu/~mweigle/papers/shankar-ms-proj-17.pdf.

[21] Al Shaw. 2011. TimelineSetter: Easy Timelines From Spreadsheets, Now Open to All. https://www.propublica.org/nerds/timelinesetter-easy-timelines-from-spreadsheets-now-open-to-all.

[22] Tim Sherratt. 2020. GLAM Workbench: Web Archives. https://glam-workbench.github.io/web-archives/.

[23] Kate Starbird. 2016. Tracing Disinformation Trajectories from the 2010 Deepwater Horizon Oil Spill. https://medium.com/hci-design-at-uw/tracing-disinformation-trajectories-from-the-2010-deepwater-horizon-oil-spill-79e8116e08f4.

[24] Kate Starbird, Dharma Dailey, Ann Hayward Walker, Thomas M. Leschine, Robert Pavia, and Ann Bostrom. 2015. Social Media, Public Participation, and the 2010 BP Deepwater Horizon Oil Spill. *Human and Ecological Risk Assessment: An International Journal* 21, 3 (2015), 605–630. https://doi.org/10.1080/10807039.2014.947866

[25] Herbert Van de Sompel, Michael L. Nelson, and Robert Sanderson. 2013. HTTP Framework for Time-Based Access to Resource States – Memento. RFC 7089.

[26] Michele C. Weigle. 2017. Visualizing Webpage Changes Over Time - new NEH Digital Humanities Advancement Grant. https://ws-dl.blogspot.com/2017/10/2017-10-16-visualizing-webpage-changes.html.

[27] Michele C. Weigle and Michael L. Nelson. 2015. Visualizing Digital Collections of Web Archives - Final Report for Columbia Libraries Web Archiving Incentive Program. http://www.cs.odu.edu/~mweigle/papers/Columbia-arcvis-final-report-2015.pdf.

[28] Chase West and Greg Franko. 2018. GifShot. https://yahoo.github.io/gifshot/.